The information provided covers key SQL statements and clauses, summarizing their usage, syntax, and examples. Here's a structured recap for quick reference:

**Core SQL Statements and Clauses**

1. **SELECT Statement**
   * **Purpose**: Retrieves data from one or more tables.
   * **Syntax**:
   * SELECT column1, column2
   * FROM table\_name;
   * **Example**:
   * SELECT first\_name, last\_name
   * FROM employees;
2. **WHERE Clause**
   * **Purpose**: Filters rows based on specified conditions.
   * **Syntax**:
   * SELECT column1, column2
   * FROM table\_name
   * WHERE condition;
   * **Example**:
   * SELECT first\_name, last\_name
   * FROM employees
   * WHERE department\_id = 90;
3. **ORDER BY Clause**
   * **Purpose**: Sorts results in ascending or descending order.
   * **Syntax**:
   * SELECT column1, column2
   * FROM table\_name
   * ORDER BY column1 [ASC|DESC];
   * **Example**:
   * SELECT first\_name, last\_name
   * FROM employees
   * ORDER BY last\_name ASC;
4. **INSERT INTO Statement**
   * **Purpose**: Adds new rows to a table.
   * **Syntax**:
   * INSERT INTO table\_name (column1, column2)
   * VALUES (value1, value2);
   * **Example**:
   * INSERT INTO departments (department\_id, department\_name)
   * VALUES (280, 'Marketing');
5. **UPDATE Statement**
   * **Purpose**: Modifies existing records in a table.
   * **Syntax**:
   * UPDATE table\_name
   * SET column1 = value1
   * WHERE condition;
   * **Example**:
   * UPDATE employees
   * SET salary = 6000
   * WHERE employee\_id = 101;
6. **DELETE Statement**
   * **Purpose**: Removes rows from a table based on conditions.
   * **Syntax**:
   * DELETE FROM table\_name
   * WHERE condition;
   * **Example**:
   * DELETE FROM employees
   * WHERE employee\_id = 102;
7. **JOIN Clause**
   * **Purpose**: Combines rows from multiple tables.
   * **Types**:
     + INNER JOIN
     + LEFT JOIN
     + RIGHT JOIN
   * **Example**:
   * SELECT e.first\_name, e.last\_name, d.department\_name
   * FROM employees e
   * INNER JOIN departments d
   * ON e.department\_id = d.department\_id;
8. **GROUP BY Clause**
   * **Purpose**: Groups rows and applies aggregate functions.
   * **Syntax**:
   * SELECT column1, COUNT(column2)
   * FROM table\_name
   * GROUP BY column1;
   * **Example**:
   * SELECT department\_id, COUNT(employee\_id)
   * FROM employees
   * GROUP BY department\_id;
9. **HAVING Clause**
   * **Purpose**: Filters grouped rows, similar to WHERE.
   * **Syntax**:
   * SELECT column1, aggregate\_function(column2)
   * FROM table\_name
   * GROUP BY column1
   * HAVING aggregate\_function(column2) condition;
   * **Example**:
   * SELECT department\_id, COUNT(employee\_id)
   * FROM employees
   * GROUP BY department\_id
   * HAVING COUNT(employee\_id) > 5;
10. **LIMIT Clause**
    * **Purpose**: Limits the number of rows in the result set.
    * **Syntax**:
    * SELECT column1, column2
    * FROM table\_name
    * LIMIT number\_of\_rows;
    * **Example**:
    * SELECT first\_name, last\_name
    * FROM employees
    * LIMIT 10;
11. **DISTINCT Clause**
    * **Purpose**: Removes duplicate rows from the result set.
    * **Syntax**:
    * SELECT DISTINCT column1
    * FROM table\_name;
    * **Example**:
    * SELECT DISTINCT department\_id
    * FROM employees;

**Usage Notes:**

* **WHERE vs HAVING**:
  + Use WHERE for filtering rows before grouping.
  + Use HAVING for filtering after applying aggregate functions.
* **ORDER BY**:
  + Default sorting is ascending (ASC).
* **LIMIT**:
  + Useful for pagination or sampling data.

Let me know if you'd like examples or further clarifications!